Edge Insertion for Optimal Triangulations[1]

M. Bern[2], H. Edelsbrunner[3], D. Eppstein[4], S. Mitchell[5] and T. S. Tan[3]

Abstract

The edge-insertion paradigm improves a triangulation of a finite point set in $\Re^2$ iteratively by adding a new edge, deleting intersecting old edges, and retriangulating the resulting two polygonal regions. After presenting an abstract view of the paradigm, this paper shows that it can be used to obtain polynomial time algorithms for several types of optimal triangulations.

Keywords. Computational geometry, optimization, two and three dimensions, point sets, triangulations, iterative improvement, edge-insertion, maxmin height, minmax slope, minmax eccentricity, polynomial time.

Appear in: *Discrete & Computational Geometry*, **10 (1)**, 47–65, (1993)

[1]Research of the second author is supported by the National Science Foundation under grant no. CCR-8921421 and under the Alan T. Waterman award, grant no. CCR-9118874. Any opinions, finding and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the view of the National Science Foundation. Part of the work was done while the second, third, and fourth authors visited the Xerox Palo Alto Research Center. The fifth author is on study leave from the National University of Singapore, Republic of Singapore.

[2]Xerox Palo Alto Research Center, 3333 Coyote Hill Road, Palo Alto, CA 94304, USA.

[3]Department of Computer Science, University of Illinois, Urbana, Illinois 61801, USA.

[4]Department of Information and Computer Science, University of California, Irvine, CA 92717, USA.

[5]Center for Applied Mathematics, Cornell University, Ithaca, NY 14853, USA.

1 Introduction

A *triangulation* of a set S of n points in $\Re^2$ is a maximally connected straight line plane graph whose vertices are the points of S. Maximality implies that all bounded faces are triangles. If the triangulation is restricted to within a connected polygonal region bounded by pairwise disjoint edges connecting points of S then it is referred to as a *constrained triangulation*. A triangulation of S can be viewed as a particular constrained triangulation where the polygonal region is bounded by the points and edges on the boundary of the convex hull of S and the other points form holes in the region. A special case of a constrained triangulation is a *polygon triangulation* where S is the set of vertices of a simple polygon and the triangulation is restricted to within the polygon.

Various criteria that can be used to define optimal triangulations arise in areas such as finite element analysis [StFi73], computational geometry [PrSh85], and surface approximation [DLR90]. Many of these criteria are defined as *maxmin* (short for maximizes the minimum) or *minmax* of some triangle or edge measure. The first quantifier is over all triangulations of the same point set and the second is over all triangles or edges of a triangulation. Two example criteria are maxmin area and maxmin inscribed circle (see [Schu87]).

The problem of automatically generating optimal triangulations for a given point set has been a subject for research since the 1960's (see e.g. the discussion in [Geor71]). Exhaustive search can be ruled out since a set of n points has, in general, exponentially many triangulations. In spite of the attention these optimization problems have received, only very little is known about constructing optimal triangulations in polynomial time. An important negative result is the NP-completeness of the following decision problem [Llo77]: given a collection of points and edges, decide whether or not there is a subset of the edges that defines a triangulation of the points. Most positive results are related to the *Delaunay triangulation* defined for finite point sets [Del34]. It has been shown that among all triangulations of a given point set, the Delaunay triangulation optimizes various criteria. These include the maxmin angle [Sib78], the minmax circumscribed circle [D'AS89], the minmax smallest enclosing circle [D'AS89, Raj91], and the minimum integral of the gradient squared [Rip90]. Efficient algorithms for constructing Delaunay triangulations are abundant in the literature and based on such diverse algorithmic paradigms as edge-flipping [Laws72, Laws77], divide-and-conquer [ShHo75, GuSt85], geometric transformation [Brow79], plane-sweep [For87], and randomized incrementation [GuKS90]. Recently, polynomial time algorithms have also been found for the minmax angle and the minmax edge length criteria [EdTW92, EdTa91].

The method of [EdTW92] is most relevant to this paper. It constructs a minmax angle triangulation by iterative application of the so-called edge-insertion operation. This paper presents an abstraction of this method, termed the *edge-insertion paradigm*, and applies it to get polynomial time algorithms for other optimal triangulation problems. Given a set of n points in $\Re^2$, the specific results are an $O(n^2 \log n)$ time algorithm that constructs a triangulation with maxmin triangle height, and $O(n^3)$ time algorithms for triangulations with minmax (three-dimensional) slope and with minmax eccentricity of any triangle. Triangulations with maxmin height have been suggested for use in surface approximation [GoCR77], and all three criteria have been mentioned in a survey article on "systematic" triangulations [WaPh84].

Section 2 formulates the most basic version of the edge-insertion paradigm, and section 3 gives two sufficient conditions for criteria it can optimize. The correctness of the paradigm when applied to such criteria is established in section 4. Section 5 discusses refinements of the paradigm and proves their correctness for the classes of criteria satisfying each of the two conditions. Sections 6, 7, and 8 demonstrate the application of the method to the three specific optimization criteria mentioned above. Section 9 concludes the paper.

2 The Edge-Insertion Paradigm

First some definitions. We denote by xy the relatively open line segment that connects the points $x, y \in \Re^2$. For $x, y, z \in \Re^2$, xyz is the open triangle with corners x, y, z. For a given finite point set $S \subseteq \Re^2$ and $x, y, z \in S$, we call xyz an *empty* triangle if all other points of S lie outside the closure of xyz.

Let μ be a function that maps a triangle xyz to a real value $\mu(xyz)$, called the *measure* of xyz. We restrict our attention to maxmin criteria, that is, for each μ we consider the construction of a triangulation that maximizes the minimum $\mu(xyz)$ over all triangles xyz. Minmax criteria can be simulated by considering $-\mu$. The measures of particular interest in this paper are the largest angle, the height, the slope, and the eccentricity of a triangle. The *measure* of a triangulation $\mathcal{A}$ is defined as $\mu(\mathcal{A}) = \min\{\mu(xyz) \mid xyz \text{ a triangle of } \mathcal{A}\}$. If $\mathcal{A}$ and $\mathcal{B}$ are two triangulations of a common point set then $\mathcal{B}$ is called an *improvement* of $\mathcal{A}$, denoted $\mathcal{A} \prec \mathcal{B}$, if $\mu(\mathcal{A}) < \mu(\mathcal{B})$ or $\mu(\mathcal{A}) = \mu(\mathcal{B}) = \mu_0$ and the set of triangles xyz in $\mathcal{B}$ with $\mu(xyz) = \mu_0$ is a proper subset of the set of such triangles in $\mathcal{A}$. A triangulation $\mathcal{T}$ is *optimal* for μ if there is no improvement of $\mathcal{T}$.

The idea of the edge-insertion paradigm is fairly simple and explained below. Its non-trivial aspects are the proof of correctness (sections 3 and 4) and the improvement of the running time from $O(n^8)$ to $O(n^3)$ and $O(n^2 \log n)$ (section 5). Given a triangulation $\mathcal{A}$ of a point set S, the *edge-insertion* of ab, $a, b \in S$, works as follows.

Function EDGE-INSERTION($\mathcal{A}$,ab): triangulation.

1. $\mathcal{B} := \mathcal{A}$.
2. Add ab to $\mathcal{B}$ and remove from $\mathcal{B}$ all edges that intersect ab.
3. Retriangulate the polygonal regions P and R constructed in step 2.
4. **return** $\mathcal{B}$.

There are many ways to triangulate the polygonal regions. For now we might as well assume that P and R are triangulated in an optimal fashion (maximizing the minimum μ), e.g. by dynamic programming [Klin80]. The basic version of the *edge-insertion paradigm* can now be formulated as follows.

Input. A set S of n points in $\Re^2$.
Output. An optimal triangulation $\mathcal{T}$ of S.

Algorithm. Construct an arbitrary triangulation $\mathcal{A}$ of S.
repeat $\mathcal{T} := \mathcal{A}$;
  for all pairs $a, b \in S$ **do**
    $\mathcal{B} :=$ EDGE-INSERTION($\mathcal{A}, ab$);
    if $\mathcal{A} \prec \mathcal{B}$ **then** $\mathcal{A} := \mathcal{B}$; **exit** the for-loop **endif**

endfor
until $\mathcal{T} = \mathcal{A}$.

The correctness of this paradigm hinges on the fact that there is an edge-insertion that improves $\mathcal{A}$, unless $\mathcal{A}$ is already optimal. Section 3 will present two conditions on criteria that can be optimized and section 4 proves that either is sufficient to imply correctness of the edge-insertion paradigm.

Assuming correctness, we argue that the above algorithm runs in time $O(n^8)$; an improvement to $O(n^3)$ and $O(n^2 \log n)$ will be given in section 5. It is reasonable to assume that a single edge-insertion operation takes time $O(n^3)$ (for retriangulating by dynamic programming). This is fair as long as the measures of any two triangles can be compared in constant time. The for-loop thus takes time $O(n^5)$ per iteration of the repeat-loop. Finally, the repeat-loop is iterated at most $O(n^3)$ times because there are only $\binom{n}{3}$ triangles spanned by S, and each iteration permanently discards at least one of them while finding an improvement of the current triangulation.

Remark. The edge-insertion paradigm can be extended to constrained triangulations by limiting the edge-insertion operation to edges ab that lie in the interior of the restricting polygonal region. As a consequence, a triangulation that lexicographically maximizes the increasing vector of triangle measures can be constructed in the non-degenerate case, that is, when $\mu(abc) \neq \mu(xyz)$ unless $abc = xyz$. Details can be found in [EdTW92].

3 Two Sufficient Conditions

We are now ready to formulate two sufficient conditions for measures μ that are amenable to the edge-insertion paradigm. Condition (I) is strictly weaker than (II), so the correctness of the paradigm needs to be established only for (I). The greater generality of (I) is off-set by a faster implementation of the edge-insertion paradigm for criteria that satisfy (II).

Let S be a set of n points in $\Re^2$, let $\mathcal{A}$ and $\mathcal{T}$ be two triangulations of S, and let xyz be a triangle in $\mathcal{A}$. We say that $\mathcal{T}$ *breaks* xyz *at* y if it contains an edge yt with $yt \cap xz \neq \emptyset$. Note that if $\mathcal{T}$ breaks xyz at y then it can neither break it at x nor at z. Conditions (I) and (II) are based on the definition of an *anchor* of a triangle xyz. Whether or not a vertex of xyz is an anchor depends solely on xyz and μ. The first condition requires that

for every triangle xyz of $\mathcal{A}$, if $\mathcal{T}$ neither contains xyz nor breaks it at its anchor(s) then $\min\{\mu(\mathcal{A}), \mu(\mathcal{T})\} < \mu(xyz)$. (I)

In other words, $\mathcal{T}$ can be an improvement of $\mathcal{A}$ only if it breaks a worst triangle of $\mathcal{A}$ at its unique anchor. The second condition requires that

for every triangle xyz of $\mathcal{A}$, if $\mathcal{T}$ neither contains xyz nor breaks it at its anchor(s) then $\mu(\mathcal{T}) < \mu(xyz)$. (II)

The important difference between the two conditions is that in (I) the triangulation $\mathcal{A}$ that contains xyz plays an important role, while in (II) $\mathcal{A}$ is insignificant. Obviously, if μ satisfies (II) then it also satisfies (I). We will see in sections 7 and 8 that the minmax slope and the minmax eccentricity criteria satisfy (I) but not (II). This shows that (I) is strictly weaker than (II).

Typically, the anchor of a triangle is unique, but this is not necessarily so. Consider for example the case where $\alpha(xyz)$ is equal to the largest angle of xyz, and $\mu(xyz) = -\alpha(xyz)$. Condition (II) is satisfied if each vertex at which the largest angle occurs is defined to be an anchor of xyz. Thus, if S defines an empty triangle xyz with two or three anchors then $\mu(\mathcal{A}) \leq \mu(xyz)$ for every triangulation $\mathcal{A}$ of S. This is because no triangulation can break xyz at more than one vertex. A special case of this statement is that if xyz is an empty isosceles triangle with two largest angles then no triangulation of S can have minmax angle less than $\alpha(xyz)$.

We will see in section 6 that $\mu(xyz)$ equal to the height of xyz satisfies (II). Section 7 considers the problem where each point $x = (\xi_1, \xi_2)$ in S has associated an "elevation" ξ_3, and a triangulation of S is "lifted" to a piecewise linear surface through the points (ξ_1, ξ_2, ξ_3). We will see that $-\mu(xyz)$ equal to the slope of the lifted triangle satisfies (I) but not (II). Define the *eccentricity* of a triangle xyz equal to the infimum distance between the center of the circumcircle and any point of xyz. Section 8 demonstrates that $-\mu(xyz)$ equal to the eccentricity of xyz also satisfies (I) but not (II).

4 The Cake Cutting Lemma Revisited

The cake cutting lemma (below) asserts that if $\mathcal{A}$ is not yet optimal then there is an edge whose insertion leads to an improvement. In [EdTW92], the cake cutting lemma is proved in the context of the minmax angle criterion using an argument that rotates edges of an optimal triangulation $\mathcal{T}$ of S. While this is appropriate for angles, we need a different argument for the more general class of measures that satisfy (I). As mentioned before, the correctness of the paradigm for (I) implies the correctness for (II). Before continuing, we remark that the regions P and R (created in step 2 of an edge-insertion) are not necessarily simple polygons in the usual meaning of the term. Although their interiors are always simply connected, there can be edges contained in the interiors of their closures. Nevertheless, each such edge can be treated as if it consisted of two edges, one for each side, which then allows us to treat P and R as if they were simple polygons.

We need some definitions. A *diagonal* of a simple polygon is a line segment that connects two vertices and lies inside the polygon. An *ear* is a triangle bounded by two polygon edges and one diagonal.

Lemma 4.1 (Cake Cutting) Let $\mathcal{A} \prec \mathcal{T}$ be two triangulations of S, let pqr be a triangle in $\mathcal{A}$ with $\mu(pqr) = \mu(\mathcal{A})$ that is not in $\mathcal{T}$, let q be an anchor of pqr, and let qs be an edge in $\mathcal{T}$ that intersects pr. Let P and R be the polygons generated by adding qs to $\mathcal{A}$ and removing all edges that intersect qs. Then there are triangulations $\mathcal{P}$ and $\mathcal{R}$ of P and R with $\mu(pqr) < \min\{\mu(\mathcal{P}), \mu(\mathcal{R})\}$.

Proof. We prove the assertion for P, and by symmetry it follows for R. The plan is to use edges of $\mathcal{T}$ as guides to successively remove ears from P to obtain $\mathcal{P}$. More specifically, we use pieces of edges of $\mathcal{T}$ that can be seen through the "window" P. Each connected component of an edge of $\mathcal{T}$ intersected with P is called a *clipped edge*. As P is not necessarily convex, several clipped edges can belong to the same edge of $\mathcal{T}$.

If no clipped edge exists in the window, then P has only three vertices and therefore must be a triangle of $\mathcal{T}$. We are done because this triangle is not in $\mathcal{A}$ which implies that its measure exceeds $\mu(\mathcal{A})$. In the following, we thus assume the existence of at least one clipped edge. Denote by $q = p_0, p_1, \ldots, p_k, p_{k+1} = s$ the sequence of vertices of P.

Claim 1. For $1 \leq j \leq k$, if $\angle p_{j-1}p_jp_{j+1} < \pi$ then $p_{j-1}p_{j+1}$ is a diagonal of P.

Proof (of Claim 1). By construction of P, it is possible to find non-intersecting line segments $p_{j-1}x$ and $p_{j+1}y$, both inside P, so that x and y lie on qs ($x = p_{j-1} = q$ if $j = 1$ and $y = p_{j+1} = s$ if $j = k$). The (possibly degenerate) pentagon $xp_{j-1}p_jp_{j+1}y$ is part of P, and because p_i, x, and y are convex vertices, the edge $p_{j-1}p_{j+1}$ is a diagonal of the pentagon and therefore also of P. This completes the proof of Claim 1.

A clipped edge partitions P into two polygons, the *near side* supported by qs and the *far side* not supported by qs.

Claim 2. There is at least one clipped edge whose far side is a triangle.

Proof (of Claim 2). Let xy be a clipped edge so that its far side, F, contains no further clipped edge. Let ab be the edge in $\mathcal{T}$ that contains xy, and let abc be the triangle in $\mathcal{T}$ that lies on the same side of xy as F. By assumption we have $F \subseteq abc$. All vertices of F, except possibly x and y, are points in S and therefore equal to a, b, or c. But unless F is also a triangle this contradicts the fact that, by construction, the angles at x and y inside F are strictly less than π. This proves Claim 2.

The clipped edges xy that satisfy Claim 2 fall into four classes as illustrated in Figure 4.1. An

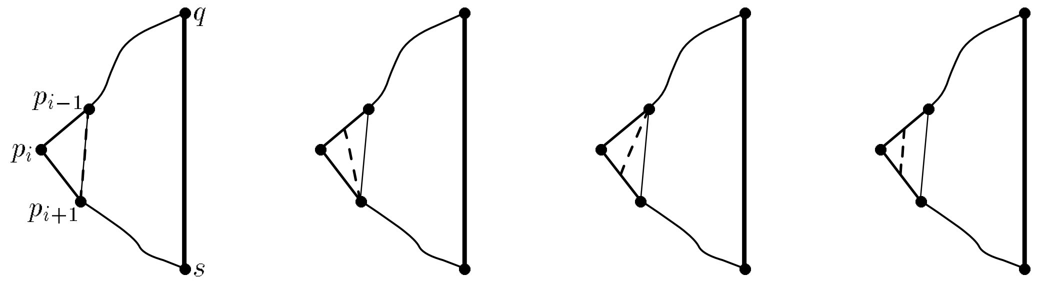

Figure 4.1: A clipped edge xy that satisfies Claim 2 has zero, one, or two endpoints on edges of P.

ear $p_{i-1}p_ip_{i+1}$ so that xy is a clipped edge with far side xp_iy can now be removed from P, leaving a polygon P' with one less vertex. Claims 1 and 2 remain true for P' because the removed ear is not supported by qs. So we can iterate and compute a triangulation $\mathcal{P}$ of P. Symmetrically, we get a triangulation $\mathcal{R}$ of R. Let $\mathcal{B}$ be the thus obtained triangulation of S.

Claim 3. $\mu(pqr) < \mu(abc)$ for all triangles abc in $\mathcal{P}$ and $\mathcal{R}$.

Proof (of Claim 3). Let abc be a triangle in $\mathcal{P}$ or $\mathcal{R}$ with minimum measure μ. Assume without loss of generality that abc is a triangle of $\mathcal{P}$ and that $a = p_i, b = p_j, c = p_k$ with $i < j < k$. At the time immediately before abc was removed by adding the edge ac there was a clipped edge xy with far side xby. Hence, $\mathcal{T}$ does not break abc at b, and by construction, $\mathcal{A}$ breaks abc at b and therefore neither at a nor at c.

If $xy = ac$ then abc is a triangle in $\mathcal{T}$ that is not in $\mathcal{A}$, and therefore $\mu(pqr) = \mu(\mathcal{A}) < \mu(abc)$. If $xy \neq ac$ and b is an anchor of abc then (I) implies $\min\{\mu(\mathcal{B}), \mu(\mathcal{T})\} < \mu(abc)$ because $\mathcal{T}$ does not break abc at b. Finally, if $xy \neq ac$ and b is not an anchor of abc then a or c is one. Because $\mathcal{A}$ does not break abc at its anchor we get $\min\{\mu(\mathcal{B}), \mu(\mathcal{A})\} < \mu(abc)$ from (I). This completes the proof of Claim 3 because all triangles of $\mathcal{B}$ are either in $\mathcal{A}$, $\mathcal{P}$ or $\mathcal{R}$, and abc is assumed to minimize μ over all triangles of $\mathcal{P}$ and $\mathcal{R}$. ◻

Using the cake cutting lemma we can now show that the algorithm, outlined as the basic version of the edge-insertion paradigm, makes progress as long as the current triangulation, $\mathcal{A}$, is not yet optimal. It suffices to show that the insertion of at least one edge is successful.

Lemma 4.2 Let $\mathcal{A}$ be a non-optimal triangulation of a finite point set S. Then there is an edge-insertion operation that improves $\mathcal{A}$.

Proof. Let $\mathcal{T}$ be an improvement of $\mathcal{A}$ and consider a triangle pqr in $\mathcal{A}$ with $\mu(pqr) = \mu(\mathcal{A})$ that is not in $\mathcal{T}$. Assuming q is an anchor of pqr, condition (I) implies that $\mathcal{T}$ contains an edge qs with $qs \cap pr \neq \emptyset$. Let P and R be the polygonal regions generated by adding qs and deleting the edges that intersect qs. The cake cutting lemma implies that there are polygon triangulations $\mathcal{P}$ and $\mathcal{R}$ of P and R with $\mu(pqr) < \min\{\mu(\mathcal{P}), \mu(\mathcal{R})\}$. ◻

Remark. Lemmas 4.1 and 4.2 remain true for constrained triangulations provided the optimization criterion satisfies (I) or (II) also in this more general setting. This is indeed the case for all criteria μ considered in this paper.

5 Refinements of the Paradigm

The refined versions of the edge-insertion paradigm differ from the basic one in two major ways. First, edge-insertions are restricted to edges qs that break a worst triangle pqr at its anchor q. More specifically, a subset of these edges qs is tried in a sequence computed as the edge-insertions fail to produce an improvement. Second, the two polygonal regions created by adding an edge qs are retriangulated by repeated ear-cutting (similar to the proof of the cake cutting lemma), rather than by dynamic programming.

The order of candidate insertion edges is not critical for criteria satisfying (I), but a careful choice of order can speed up the algorithm for criteria satisfying (II). Let, for example, $\mathcal{A}$ be a triangulation with worst triangle pqr, that is, $\mu(pqr) = \mu(\mathcal{A})$, and let q be its anchor. We denote by $qs_1, qs_2, \ldots, qs_j$ the sequence of edges inserted with the goal to find an improvement of $\mathcal{A}$. We will consider two refinements of the algorithm in section 2, one for each class of criteria, which differ in the sequence of edge-insertions. Both are specializations of the algorithm given below in pseudo-code. We use the notation $s_{i+1} = \text{NEXT}(s_i)$.

Algorithm. Construct an arbitrary triangulation $\mathcal{A}$ of S.
repeat $\mathcal{T} := \mathcal{A}$;
find a worst triangle pqr in $\mathcal{A}$, let q be its anchor, and set $s := s_1$;
while s is defined **do**

$\mathcal{B} := \mathcal{A}$, add qs to $\mathcal{B}$, and remove all edges that intersect qs;
(partially) triangulate the thus created two polygonal regions P
and R by cutting off ears xyz with $\mu(xyz) > \mu(pqr)$;
if P and R are completely triangulated **then**
$\mathcal{A} := \mathcal{B}$; **exit** the while-loop
else $s :=$ NEXT(s)
endif
endwhile
until $\mathcal{T} = \mathcal{A}$.

We remark that this algorithm finds a triangulation with maxmin triangle measure, but not necessarily an optimal triangulation in the sense that the set of worst triangles is minimal. To achieve this slightly more ambitious goal *all* worst triangles need to be subject to edge-insertions before the algorithm halts.

In an implementation of the algorithm we would of course not copy entire triangulations. The only reason for assigning $\mathcal{A}$ to $\mathcal{T}$ is to be able to check whether an iteration of the repeat-loop in fact produces an improved triangulation. Alternatively, this can be monitored by setting a flag whenever the first branch of the if-statement is entered. The assignment $\mathcal{B} := \mathcal{A}$ can be avoided by making changes directly in $\mathcal{A}$ and undoing them to the extent necessary. The remainder of this section explains some of the steps in greater detail and assesses the complexity of the two algorithms obtained.

Triangulating by ear cutting. Suppose an edge qs has been added to $\mathcal{B}$ and the edges that intersect qs have been removed, thus creating two regions P and R. Let $q = p_0, p_1, \ldots, p_k, p_{k+1} = s$ be the sequence of vertices of P and let $q = r_0, r_1, \ldots, r_m, r_{m+1} = s$ be the corresponding sequence for R, as shown in Figure 5.1. Mimicking the proof of the cake cutting lemma, the two regions are

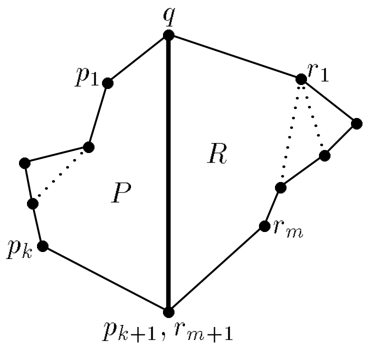

Figure 5.1: The polygons P and R are created by adding qs to $\mathcal{B}$ and removing intersecting edges. The dotted lines indicate ears with better measure than pqr.

(partially) triangulated by repeatedly removing ears with measures exceeding $\mu(pqr)$. As implied by the proof, the sequence in which the ears are removed is immaterial as long as they are not supported by qs (only the ear removed last is supported by qs). It is thus fairly straightforward to implement this method using a stack for the vertices of P (R) so that it runs in time linear in the size of P (R). In the case of P, the stack is initialized by pushing p_0 and p_1. After that, for $i := 2$ to $k+1$ we push vertex p_i, and as long as the topmost three vertices, $z = p_i, y, x$, define a triangle with $\mu(xyz) > \mu(pqr)$ we pop y, the second vertex from the top. The triangulation is complete if, at the end of the process, $p_{k+1} = s$ and $p_0 = q$ are the only two vertices on the stack.

Analysis under (I). If the insertion of an edge qs is unsuccessful, that is, the triangulation of P or that of R cannot be completed, then we know by the cake cutting lemma that qs cannot be in any improvement of the current triangulation. We record this information by setting a flag in an n-by-n bit array whose elements correspond to the edges defined by S. This way we avoid attempting the insertion of qs at any later stage of the algorithm. If the insertion of qs is successful then pr is deleted from the current triangulation, and because of condition (I) it cannot be in any later improvement. We thus set the flag for pr. The bit array can also be used to compute the sequence of edges qs_i: scan the row corresponding to q and take all edges qs that intersect pr and whose flag has not yet been set.

Theorem 5.1 Let S be a set of n points in $\Re^2$ and let μ be a measure that satisfies (I).
(1) A constrained or unconstrained triangulation of S that maximizes the minimum triangle measure can be constructed in time $O(n^3)$ and storage $O(n^2)$.
(2) In the non-degenerate case (i.e. when $\mu(xyz) \neq \mu(abc)$ unless $xyz = abc$) the (unique) triangulation that lexicographically maximizes the increasing vector of triangle measures can be constructed in the same amount of time and storage.

Proof. To achieve the claimed bounds, the above algorithm uses two data structures taking a total of $O(n^2)$ storage. First, the quad-edge data structure of Guibas and Stolfi [GuSt85] stores the triangulation in $O(n)$ memory and admits common operations, such as removing an edge, adding an edge, and walking from one edge to the next in constant time each. The other data structure is the bit array mentioned above. The quad-edge data structure together with the ear cutting method explained above allows an edge-insertion to be completed in time $O(n)$ as only a linear number of edges have to be removed and added. Each edge-insertion, whether successful or not, causes a new flag set for one of the $\binom{n}{2}$ edges defined by S. Therefore, at most $\binom{n}{2}$ edge-insertions are carried out taking a total of $O(n^3)$ time. Part (1) of the claim follows because an initial triangulation can be constructed in time $O(n \log n)$, most straightforwardly by plane-sweep (see [Edel87, section 8.3.1]).

To get a triangulation that lexicographically maximizes the entire vector of triangle measures we solve a sequence of constrained triangulation problems as in [EdTW92]. The first constraining region is defined by the points and edges on the boundary of the convex hull of S with the other points forming holes. After computing an optimal triangulation as in (1), we remove the worst triangle (which is unique by non-degeneracy assumption) from the constraining region and iterate until the region is empty. The time is still $O(n^3)$ because each edge needs to be inserted at most once during the entire process. ◻

Searching for the right edge. For measures μ that satisfy (II) we can be more clever about the sequence $qs_1, qs_2, \ldots, qs_j$ of edge-insertions. The first edge, qs_1, has the property that it intersects pr, but otherwise it intersects as few edges as possible. If s_1 exists then it is unique. If s_1 does not exist then $j = 0$, that is, no edge-insertion is attempted. As discussed below, every qs_{i+1} has the property that s_i lies on a particular side of qs_{i+1}, and with this constraint the set of edges in $\mathcal{B}$ that intersect qs_{i+1} is the smallest proper superset of the set of edges that intersect qs_i. The index j is the smallest integer for which qs_j leads to an improvement or s_{j+1} is undefined.

The retriangulation process either completes its task or it gets stuck because all ears of the remaining regions that are not supported by qs have measure less than or equal to $\mu(pqr)$. Combining the different cases for P and R we get four possible outcomes. If P and R are both completely triangulated then an improvement of $\mathcal{B}$ has been obtained and the algorithm exits the while-loop. Let us now consider the case where the triangulation of P cannot be completed. In this case, the stack contains $k+2 \geq 3$ vertices $q = p_0, p_1, \ldots, p_k, p_{k+1} = s$ defining the remaining region $P' \subseteq P$; each ear $p_{i-1}p_ip_{i+1}$ of P' has measure at most $\mu(pqr)$. The next lemma is crucial for defining $\text{NEXT}(s)$.

Lemma 5.2 Let $\mathcal{T}$ be an improvement of $\mathcal{B}$. Then all edges of $\mathcal{T}$ that intersect P' also intersect qs. In particular, all edges of $\mathcal{T}$ incident to q avoid P'.

Proof. As in the proof of the cake cutting lemma we consider P' as a "window" through which we see clipped edges of $\mathcal{T}$. Now suppose the claim is not true, that is, there is a clipped edge that does not have one of its endpoints on qs. As before we thus find such a clipped edge xy whose far side is a triangle xp_iy. But now condition (II) implies $\mu(\mathcal{T}) < \mu(p_{i-1}p_ip_{i+1})$ if p_i is an anchor of the ear $p_{i-1}p_ip_{i+1}$, and $\mu(\mathcal{B}) < \mu(p_{i-1}p_ip_{i+1})$ if p_{i-1} or p_{i+1} are anchors. This contradicts the assumption that P' has no such ear. $\square$

It is interesting to observe that the proof of Lemma 5.2 breaks down if we only assume that μ satisfies (I) but not (II).

Lemma 5.2 suggests that we maintain an open wedge W where all points s must lie for which qs is possibly a successful edge-insertion. Initially, W is the wedge between the half-line $\vec{qp}$ (it starts at q and passes through p) and the half-line $\vec{qr}$. If the edge-insertion of qs turns out to be unsuccessful because the triangulation of P cannot be completed then W can be redefined as the part of the old W on R's side of $\vec{qs}$. Similarly, if the triangulation of R cannot be completed then W can be narrowed down to P's side of $\vec{qs}$. As a consequence, if neither P nor R can be completely triangulated then it is impossible to improve the current triangulation by breaking pqr at q. For reasons that will become clear shortly, it is however too costly to check when this is the case. As soon as one polygon has been found to be non-completable, the wedge is updated and an edge-insertion is attempted with the next point s.

When we move from qs_i to qs_{i+1}, most of the work done to triangulate P and R can be saved. Assume that qs_i has been abandoned because P could not be completely triangulated. Because qs_{i+1} intersects r_mr_{m+1} (the last edge of R) and thus moves away from P, all ears cut off P are fine and do not have to be reconsidered. On the other hand, r_{m+1} is no longer a vertex of R, so all ears cut off R that are incident to r_{m+1} must be returned to R's territory. When we move to qs_{i+1} some additional edges are removed from $\mathcal{B}$ which, in effect, expands P and R. The new vertices of P can just be pushed on P's stack, one by one, so that the triangulation process can continue where it stopped. Similar for R.

The only place where time is wasted when we move from qs_i to qs_{i+1} is when ears cut off one polygon (in the above discussion this is R) are returned to this polygon. Since ears are returned only for one polygon we can limit the waste by strictly alternating between cutting an ear of P and one of R. This way, for each but possibly one recycled ear there is a permanently removed ear. Therefore, the total number of operations performed while edge-inserting $qs_1, qs_2, \ldots, qs_j$ is linear in the number of edges in $\mathcal{B}$ that intersect qs_j.

Analysis under (II). As already mentioned, a successful edge-insertion, complete with retriangulation, takes a number of operations that is linear in the number of old edges intersected by the new edge. We now prove that the old edges removed will never be reinserted in any later successful edge-insertion.

Lemma 5.3 Let $\mathcal{A}$ be a triangulation of S, with worst triangle pqr, and let $\mathcal{B}$ be obtained from $\mathcal{A}$ by the successful insertion of an edge qs. Then no edge xy in $\mathcal{A}$ that intersects qs can be an edge of any improvement of $\mathcal{B}$.

Proof. Lemma 5.2 implies that every improvement of $\mathcal{B}$ has an edge qw that lies inside the wedge W computed when qs is inserted into $\mathcal{A}$. Every edge xy in $\mathcal{A}$ that intersects qs also intersects every other edge qt with $t \in W$. In particular, $xy \cap qw \neq \emptyset$ which implies that xy is neither in $\mathcal{B}$ nor in any improvement of $\mathcal{B}$. □

Now we have all the ingredients to repeat the time-analysis of [EdTW92] in the context of criteria that satisfy condition (II).

Theorem 5.4 Let S be a set of n points in $\Re^2$ and let μ be a measure that satisfies (II).
(1) A constrained or unconstrained triangulation of S that maximizes the minimum triangle measure can be constructed in time $\mathrm{O}(n^2 \log n)$ and storage $\mathrm{O}(n)$.
(2) In the non-degenerate case (i.e. when $\mu(xyz) \neq \mu(abc)$ unless $xyz = abc$) the (unique) triangulation that lexicographically maximizes the increasing vector of triangle measures can be constructed in the same amount of time and storage.

Proof. As before, the algorithm uses the quad-edge data structure of [GuSt85] to store the triangulation. The second data structure is a priority queue that holds the triangles of $\mathcal{A}$ ordered by measure. It admits inserting and deleting a triangle and finding a triangle with minimum measure in logarithmic time each [CLR90]. Lemma 5.3 implies that only $\mathrm{O}(n^2)$ edges and triangles are manipulated in the main loop of the algorithm, which thus takes time $\mathrm{O}(n^2 \log n)$, a logarithmic share per edge to cover the expenses for the priority queue operations. Lemma 5.3 also implies a quadratic upper bound on the number of iterations of the repeat-loop, which implies that the total time needed to find worst triangles pqr is also $\mathrm{O}(n^2 \log n)$. This proves part (1), and (2) follows by the same argument as in Theorem 5.1. □

6 Maximizing the Minimum Height

For a (finite) point set S, a *maxmin height triangulation* of S maximizes the smallest height of its triangles, over all triangulations of S. For a triangle there are three ways to define a *base edge* zx and an *apex* y. Let $h(y, zx)$ be the minimum distance between y and a point on the line through z and x. Then the *height* (or *width*) of a triangle xyz is defined as $\eta(xyz) = \min\{h(x, yz), h(y, zx), h(z, xy)\}$. It is easy to see that $h(y, zx) < h(z, xy)$ iff $\angle xyz > \angle yzx$. Therefore, $\eta(xyz) = h(y, zx)$ iff the angle at y is at least as large as the angles at x and z.

Although the maxmin height, the maxmin angle, and the minmax angle criteria all tend to avoid thin and elongated triangles in the resulting optimal triangulations, they do not necessarily define the same optima. Indeed, four-point examples can be constructed to show that the three criteria are pairwise different.

The edge-flipping strategy [Laws72, Laws77] applied to the maxmin height criterion does not always succeed in computing an optimal triangulation. For consider a regular pentagon *abcde* and the circle through the five points. Perturb a slightly to a point outside the circle and c and d slightly to points inside the circle so that $h(c, db) < h(d, ec) < h(b, ca) = h(e, ad) < h(a, be)$ (see Figure 6.1). The optimal triangulation in terms of η is defined by the diagonals ac and ad. Now, if be and ce are in the current triangulation no edge-flip can result in a better triangulation.

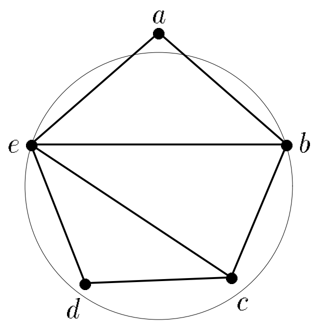

Figure 6.1: Flipping be or ce of the current triangulation both locally decreases the minimum height. Thus, the edge-flip method cannot change the shown triangulation into the optimal one.

We now show that η satisfies condition (II). It follows that maxmin height triangulations can be constructed by the $O(n^2 \log n)$ time implementation of the edge-insertion paradigm.

Lemma 6.1 Let xyz be a triangle of a triangulation $\mathcal{A}$ of S and let $\eta(xyz) = h(y, zx)$. Then $\eta(\mathcal{T}) < \eta(xyz)$ for any triangulation $\mathcal{T}$ of S that neither contains xyz nor breaks xyz at y.

Proof. The height $\eta(xyz) = h(y, zx)$ is the distance between y and a point $s \in zx$. Assume that xyz is not in $\mathcal{T}$ and that $\mathcal{T}$ does not break xyz at y. Therefore, there exists a triangle uyv in $\mathcal{T}$ so that either $u = x$ and $uv \cap yz \neq \emptyset$ (rename vertices if necessary), or uv intersects both yx and yz. In both cases, $\eta(uyv) \leq h(y, uv) < \eta(xyz)$ because $uv \cap ys \neq \emptyset$. □

For η it is thus appropriate to call y an *anchor* of xyz iff $\eta(xyz) = h(y, zx)$. It should be clear that Lemma 6.1 also holds for constrained triangulations of S. Using Theorem 5.4 we can therefore conclude that a maxmin height triangulation, and in the non-degenerate case a maxmin height vector triangulation, can be computed in time $O(n^2 \log n)$ and storage $O(n)$.

7 Minimizing the Maximum Slope

Consider a function $f : \Re^2 \to \Re$ defining a surface $x_3 = f(x_1, x_2)$ in $\Re^3$. The *gradient* of f is the vector $\nabla f = (\frac{\partial f}{\partial x_1} \frac{\partial f}{\partial x_2})$, each component of which is itself a function from $\Re^2$ to $\Re$. Define

$\nabla^2 f = (\frac{\partial f}{\partial x_1})^2 + (\frac{\partial f}{\partial x_2})^2$, and call $\sqrt{\nabla^2 f}$ at a point (x_1, x_2) the *slope* at this point.

Let $S = \{p_i = (\pi_{i1}, \pi_{i2}) \mid 1 \leq i \leq n\}$ be a point set in $\Re^2$ and let $\hat{S} = \{\hat{p}_i = (\pi_{i1}, \pi_{i2}, \pi_{i3})\}$ be the corresponding set in $\Re^3$ where each p_i has a third coordinate π_{i3}, its *elevation*. Analogous to the definitions in $\Re^2$, $\hat{x}\hat{y}$ denotes the relatively open line segment with endpoints $\hat{x}$ and $\hat{y}$, and $\hat{x}\hat{y}\hat{z}$ denotes the relatively open triangle with corners $\hat{x}, \hat{y}, \hat{z}$. We can think of $\hat{x}\hat{y}\hat{z}$ as a function f defined within xyz. For each point $w = (\omega_1, \omega_2) \in xyz$ the gradient is well defined and the same as for any other point in xyz. We can therefore set $\sigma(xyz)$ equal to the slope at w and call it the *slope* of xyz. For a triangulation $\mathcal{A}$ of S define $\sigma(\mathcal{A}) = \max\{\sigma(xyz) \mid xyz \text{ a triangle of } \mathcal{A}\}$, as usual. A *minmax slope triangulation* of S minimizes the maximum σ of any triangle.

The five point example of Figure 6.1 can also be used to argue that the edge-flipping strategy does not always succeed in computing minmax slope triangulations. Just imagine that points a, b, c, d, e are not perturbed and thus form a regular pentagon. Let the elevations of a, b, c, d, e be $5, 11, 0, 10, 0$, in this sequence. The optimal triangulation is defined by the diagonals ac and ad, and the current triangulation (with diagonals be and ce as shown) cannot be improved by a single edge-flip. The remainder of this section shows that $\mu = -\sigma$ satisfies condition (I).

Observe first that the direction of steepest descent of a triangle xyz is given by $\Delta = -\nabla f$ at a point in xyz. We call the vertex y an *anchor* of xyz unless the line $y + \lambda\Delta$, $\lambda \in \Re$, intersects the closure of xyz only in y. In the non-degenerate case xyz has only one anchor, but if Δ is parallel to an edge then there are two anchors. We will see shortly that this definition of anchor is exactly what is needed to prove that $\mu = -\sigma$ satisfies (I). Call the intersection of the closure of $\hat{x}\hat{y}\hat{z}$ with the plane parallel to the x_3-axis through $y + \lambda\Delta$ the *descent line* $\ell(xyz)$ of xyz, assuming y is an anchor of xyz.

For technical reasons it is necessary to assume that no four points of S are coplanar. Indeed, the strict inequality in Lemma 7.1 is incorrect without this assumption. This general position assumption, however, does not diminish the generality of our algorithm, because a simulated perturbation of the points can be used to simulate it [EdMü90]. This perturbation is infinitesimal. Consider the triangulation of the unperturbed points that corresponds to an optimal triangulation of the perturbed points. This triangulation must minimize the maximum slope over all triangulations of the unperturbed points.

Lemma 7.1 Let xyz be a triangle of a triangulation $\mathcal{A}$ of S and let y be an anchor of xyz. Then $\max\{\sigma(\mathcal{A}), \sigma(\mathcal{T})\} > \sigma(xyz)$ for every triangulation $\mathcal{T}$ of S that neither contains xyz nor breaks xyz at y.

Proof. The slope of xyz, $\sigma(xyz)$, is also the slope of the descent line $\ell_1 = \ell(xyz)$, see Figure 7.1. Assume without loss of generality that ℓ_1 descends from $\hat{y}$ down to where it meets the closure of $\hat{x}\hat{z}$. Assume also that $\mathcal{T}$ neither contains xyz nor breaks it at y. It follows that $\mathcal{T}$ contains an edge uv so that either $u = x$ and $uv \cap yz \neq \emptyset$ (rename vertices if necessary), or uv intersects both yx and yz. If $\sigma(uyv) > \sigma(xyz)$ then $\sigma(\mathcal{T}) > \sigma(xyz)$ and there is nothing to prove.

Otherwise, the edge $\hat{u}\hat{v}$ must pass *above* ℓ_1 in $\Re^3$. By this we mean that there is a line parallel to the x_3-axis that meets $\hat{u}\hat{v}$ and ℓ_1 and the elevation of its intersection with $\hat{u}\hat{v}$ exceeds the elevation of its intersection with ℓ_1, see Figure 7.1. Then at least one of $\hat{u}$ and $\hat{v}$ must lie above the plane

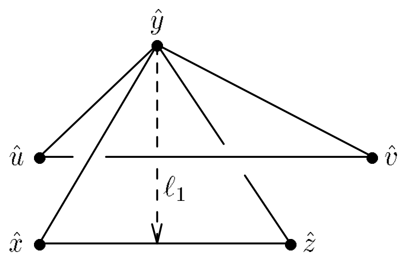

Figure 7.1: The triangle xyz with anchor y in $\mathcal{A}$ is neither contained in $\mathcal{T}$ nor is it broken at y by $\mathcal{T}$. Therefore, $\mathcal{T}$ contains a triangle uyv that intersects xyz as shown. It is possible that $u = x$ or $v = z$, but not both at the same time.

$h_1 = h(xyz)$ through points $\hat{x}, \hat{y}, \hat{z}$; say $\hat{v}$ lies above h_1. Consider the triangle yvz, and note that it is not necessarily an empty triangle of S. We have $\sigma(yvz) > \sigma(xyz)$ because the projection along x_3-direction of ℓ_1 onto the plane $h_2 = h(yvz)$ is steeper than ℓ_1 but not steeper than $\ell_2 = \ell(yvz)$. We distinguish three cases depending on which vertex is the anchor of yvz.

Case 1. v is anchor of yvz. Then ℓ_2 connects $\hat{v}$ with a point on the closure of $\hat{y}\hat{z}$. Since yz is an edge in $\mathcal{A}$ at least one of the triangles abc in $\mathcal{A}$ that intersect the projection of ℓ_2 has $\sigma(abc) \geq \sigma(yvz) > \sigma(xyz)$. This implies $\sigma(\mathcal{A}) > \sigma(xyz)$.

Case 2. z is anchor of yvz. Then ℓ_2 connects $\hat{z}$ with a point on the closure of $\hat{y}\hat{v}$. Since yv is an edge in $\mathcal{T}$ at least one of the triangles abc in $\mathcal{T}$ that intersect the vertical projection of ℓ_2 has $\sigma(abc) \geq \sigma(yvz) > \sigma(xyz)$, and therefore $\sigma(\mathcal{T}) > \sigma(xyz)$.

Case 3. y is anchor of yvz. In this case ℓ_2 connects $\hat{y}$ with a point $\hat{w}$ on the closure of $\hat{v}\hat{z}$. Furthermore, it is impossible that ℓ_2 descends from $\hat{y}$ to $\hat{w}$ because $\hat{w}$ lies above h_1, which contradicts $\sigma(yvz) > \sigma(xyz)$. Thus, it must be that ℓ_2 descends from $\hat{w}$ down to $\hat{y}$. But then $\sigma(uyv) > \sigma(yvz)$ because $\hat{u}\hat{v}$ passes above ℓ_2, a contradiction. □

Note that Lemma 7.1 also holds for constrained triangulations of S. We can therefore apply Theorem 5.1 and get an $O(n^3)$ time and $O(n^2)$ storage algorithm for constructing a minmax slope triangulation, and in the non-degenerate case for constructing a minmax slope vector triangulation.

Remark. It is interesting to observe that σ does not satisfy (II), so an $O(n^2 \log n)$ time algorithm for minmax slope triangulations seems out of reach at this moment. The example that shows that σ indeed violates (II) consists of five points with elevations as shown in Figure 7.2.

8 Minimizing the Maximum Eccentricity

Consider a triangle xyz and let (c_1, ρ_1) be its circumcircle, with center c_1 and radius ρ_1. Recall from section 3 that the *eccentricity* of xyz, $\epsilon(xyz)$, is the infimum over all distances between c_1 and points of xyz. Clearly, $\epsilon(xyz) = 0$ iff c_1 lies in xyz or on one of its edges. Note that eccentricity is related to the size of the maximum angle, $\alpha(xyz)$. Specifically,

$$\alpha(xyz) < \alpha(abc) \quad \text{iff} \quad \frac{\epsilon(xyz)}{\rho_1} < \frac{\epsilon(abc)}{\rho_2}$$

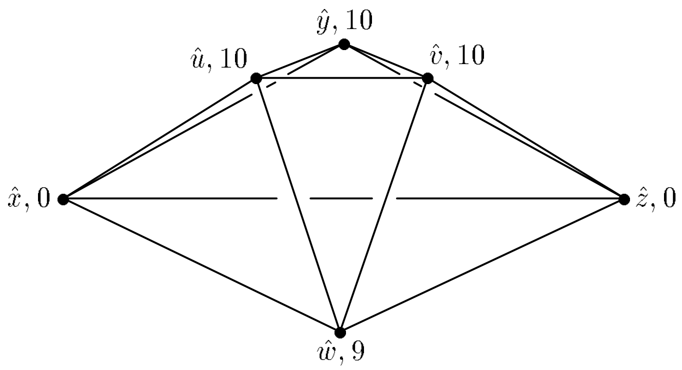

Figure 7.2: The triangulation $\mathcal{T}$ with diagonals vx and vw is an improvement of $\mathcal{A}$ with diagonals zx and zy. Consider xyz with anchor y in $\mathcal{A}$. Although $\mathcal{T}$ does not break xyz at y it still has no triangle with slope worse than xyz.

unless $\epsilon(xyz) = \epsilon(abc) = 0$, where ρ_2 is the radius of the circumcircle of abc. This suggests we call y an *anchor* of xyz if the angle at y is at least as large as the angles at x and z. As usual, we define $\epsilon(\mathcal{A})$ equal to the maximum $\epsilon(xyz)$ of any triangle xyz of $\mathcal{A}$. A *minmax eccentricity triangulation* $\mathcal{T}$ of S minimizes $\epsilon(\mathcal{A})$ over all triangulations $\mathcal{A}$ of S.

The triangulation of the pentagon in Figure 6.1 can be used to show that edge-flipping does not always succeed in minimizing the maximum eccentricity. Similarly, we can verify that $\mu = -\epsilon$ does not satisfy condition (II) by looking at Figure 7.2 and ignoring the elevation of the points. The maximum angle of xyv can be made almost as small as $\alpha(xyz)$ by moving v closer to yz. Still, the circumcircle of xyv remains significantly smaller than the one of xyz. This implies that $\epsilon(xyv)$ is smaller than $\epsilon(xyz)$. More generally, $\epsilon(xyz)$ exceeds the eccentricity of every triangle of $\mathcal{T}$, even though $\mathcal{T}$ does not break xyz at its anchor, y. Nevertheless, $\mu = -\epsilon$ satisfies condition (I).

Lemma 8.1 Let xyz be a triangle of a triangulation $\mathcal{A}$ of S and let y be an anchor of xyz. Then $\max\{\epsilon(\mathcal{A}), \epsilon(\mathcal{T})\} > \epsilon(xyz)$ for every triangulation $\mathcal{T}$ of S that neither contains xyz nor breaks xyz at y.

Proof. Assume that $\mathcal{T}$ neither contains xyz nor breaks it at y. Therefore, $\mathcal{T}$ must contain a triangle uyv so that $u = x$ and $uv \cap yz \neq \emptyset$ (rename vertices if necessary), or uv intersects yx and yz, as in Figure 7.1. Let (c_1, ρ_1) be the circumcircle of xyz. If neither u nor v are enclosed by this circle then $\epsilon(xyz) < \epsilon(uyv) \leq \epsilon(\mathcal{T})$. Otherwise, assume that v is enclosed by (c_1, ρ_1) and consider the line segment c_1v. It intersects a sequence of edges of $\mathcal{A}$, ordered from c_1 to v. For an edge ab in this sequence let abc be the supporting triangle so that c and c_1 lie on different sides of ab. Assume that ab is the first edge in the sequence so that (c_1, ρ_1) encloses c but not a and not b. Then $\epsilon(\mathcal{A}) \geq \epsilon(abc) > \epsilon(xyz)$. $\square$

Theorem 5.1 thus implies that a minmax eccentricity triangulation for n points can be constructed in time $O(n^3)$ and storage $O(n^2)$. In the non-degenerate case, the same amount of time and storage suffice to construct a minmax eccentricity vector triangulation.

9 Conclusion

The main result of this paper is the formulation of the edge-insertion paradigm as a general method to compute optimal triangulations, and the identification of classes of criteria for which the paradigm indeed finds the optimum. The paradigm is an abstraction of the algorithm introduced in [EdTW92] for computing minmax angle triangulations.

Though simple to be verified, conditions (I) and (II) are somewhat restrictive. It would be interesting to find conditions weaker than (I) even though the price to pay may be implementations of the paradigm that take more than cubic time. Listings of optimality criteria can be found in [Barn77, Lind83, Schu87]. Furthermore, implementations for criteria satisfying (I) and (II) that run in time $o(n^3)$ and $o(n^2 \log n)$ are sought.

References

[Barn77] R. E. Barnhill. Representation and approximation of surfaces. *Math. Software III*, J. R. Rice, ed., Academic Press, 1977, 69–120.

[Brow79] K. Q. Brown. Voronoi diagrams from convex hulls. *Inform. Process. Lett.* **9** (1979), 223–228.

[CLR90] T. H. Cormen, C. E. Leiserson and R. L. Rivest. *Introduction to Algorithms.* The MIT Press, Cambridge, Mass., 1990.

[D'AS89] E. F. D'Azevedo and R. B. Simpson. On optimal interpolation triangle incidences. *SIAM J. Sci. Stat. Comput.* **10** (1989), 1063–1075.

[Del34] B. Delaunay. Sur la sphère vide. *Izv. Akad. Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk* **7** (1934), 793–800.

[DLR90] N. Dyn, D. Levin and S. Rippa. Data dependent triangulations for piecewise linear interpolation. *IMA J. Numer. Anal.* **10** (1990), 137–154.

[Edel87] H. Edelsbrunner. *Algorithms in Combinatorial Geometry.* Springer-Verlag, Heidelberg, Germany, 1987.

[EdMü90] H. Edelsbrunner and E. P. Mücke. Simulation of Simplicity: a technique to cope with degenerate cases in geometric algorithms. *ACM Trans. Graphics* **9** (1990), 66–104.

[EdTa91] H. Edelsbrunner and T. S. Tan. A quadratic time algorithm for the minmax length triangulation. *In* "Proc. 32nd IEEE Sympos. Found. Comput. Sci. 1991", 414–423.

[EdTW92] H. Edelsbrunner, T. S. Tan and R. Waupotitsch. An $O(n^2 \log n)$ time algorithm for the minmax angle triangulation. *SIAM J. Stat. Sci. Comput.* **13** (1992), 994-1008.

[For87] S. Fortune. A sweepline algorithm for Voronoi diagrams. *Algorithmica* **2** (1987), 153–174.

[Geor71] J. A. George. Computer implementation of the finite element method. Techn. Rep. STAN-CS-71-208, Ph.D. Thesis, Comput. Sci. Dept., Stanford Univ., 1971.

[GoCR77] C. M. Gold, T. D. Charters and J. Ramsden. Automated contour mapping using triangular element data structures and an interpolant over each irregular triangular domain. *In* "Proc. SIGGRAPH, 1977" **11** (1977), 170–175.

[GuKS90] L. J. Guibas, D. E. Knuth and M. Sharir. Randomized incremental construction of Delaunay and Voronoi diagrams. *In* "Proc. Internat. Colloq. Automata, Lang., Progr. 1990", 414–431.

[GuSt85] L. J. Guibas and J. Stolfi. Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams. *ACM Trans. Graphics* **4** (1985), 74–123.

[Klin80] G. T. Klincsek. Minimal triangulations of polygonal domains. *Annals Discrete Math.* **9** (1980), 121–123.

[Laws72] C. L. Lawson. Generation of a triangular grid with applications to contour plotting. Jet Propul. Lab. Techn. Memo. 299, 1972.

[Laws77] C. L. Lawson. Software for C^1 surface interpolation. In *Math. Software III*, J. R. Rice, ed., Academic Press, 1977, 161–194.

[Lind83] D. A. Lindholm. Automatic triangular mesh generation on surfaces of polyhedra. *IEEE Trans. Magnetics* **MAG-19** (1983), 2539–2542.

[Llo77] E. L. Lloyd. On triangulations of a set of points in the plane. *In* "Proc. 18th Ann. IEEE Sympos. Found. Comput. Sci., 1977", 228–240.

[PrSh85] F. P. Preparata and M. I. Shamos. *Computational Geometry – an Introduction.* Springer-Verlag, New York, 1985.

[Raj91] V. T. Rajan. Optimality of the Delaunay triangulation in $\Re^d$. *In* "Proc. 7th Ann. Sympos. Comput. Geom., 1991", 357–363.

[Rip90] S. Rippa. Minimal roughness property of the Delaunay triangulation. *Computer Aided Geometric Design* **7** (1990), 489–497.

[Schu87] L. L. Schumaker. Triangulation methods. *Topics in Multivariate Approximation*, C. K. Chui, L. L. Schumaker and F. I. Utreras, eds., Academic Press, 1987, 219–232.

[ShHo75] M. I. Shamos and D. Hoey. Closest point problems. *In* "Proc. 16th Ann. IEEE Sympos. Found. Comput. Sci., 1975", 151–162.

[Sib78] R. Sibson. Locally equiangular triangulations. *Comput. J.* **21** (1978), 243–245.

[StFi73] G. Strang and G. Fix. *An Analysis of the Finite Element Method.* Prentice-Hall, Englewood Cliffs, NJ, 1973.

[WaPh84] D. F. Watson and G. M. Philip. Systematic triangulations. *Comput. Vision, Graphics, Image Process.* **26** (1984), 217–223.